

### REMARKS

As may be appreciated from the listing of claims provided above, the claims have been amended herein. Authorization is provided herewith to pay any underpayment of fees or credit any overpayment of fees to Deposit Account No. 02-4800.

#### **I. RESPONSE TO THE REJECTION OF CLAIMS UNDER 35 U.S.C. § 112**

In the Office Action dated February 4, 2009 (hereafter "Office Action"), claims 11, 13 and 19 were rejected under 35 U.S.C. § 112, Second Paragraph, as being indefinite. (Office Action, at 2). Specifically, the Examiner stated that the limitation "the group" in the second line of claims 11, 13 and 19 did not have sufficient antecedent basis. (*Id.*).

Applicant refers the Examiner to MPEP § 2173.05(h), which states

Alternative expressions are permitted if they present no uncertainty or ambiguity with respect to the question of scope or clarity of the claims. **One acceptable form of alternative expression**, which is commonly referred to as a Markush group, recites members as being **"selected from the group** consisting of A, B and C." *See Ex parte Markush*, 1925 C.D. 126 (Comm'r Pat. 1925). (emphasis added).

Original claims 11, 13 and 19 have sufficient antecedent basis. These claims are drafted in a Markush group format. As clearly described in MPEP § 2173.05(h), such a format is not indefinite and is, in fact, acceptable. Therefore, claims 11, 13 and 19 have proper antecedent basis. The rejection of claims 11, 13 and 19 based on 35 U.S.C. § 112 should be withdrawn.

#### **II. RESPONSE TO THE REJECTION OF THE CLAIMS UNDER 35 U.S.C. § 103**

The Examiner rejected pending claims 9-12, 14-19 and 21-23 under 35 U.S.C. § 103 in view of the combination of U.S. Patent No. 7,069,517 to Washington et al. and U.S. Patent No. 6,642,942 to Crook in the Office Action. (Office Action, at 2-7). Pending claims 13 and 20

were rejected as obvious based on the combination of Washington et al. with Crook and U.S. Patent No. 7,068,299 to Lemieux et al. (Office Action, at 8).

**A. Burden Of Proving Obviousness Under 35 U.S.C. § 103**

**"All words in a claim must be considered in judging the patentability of that claim against the prior art."** MPEP § 2143.03 (emphasis added). "When evaluating claims for obviousness under 35 U.S.C. 103, **all the limitations of the claims must be considered and given weight.**" MPEP § 2143.03. "If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious." *Id.* "A 35 U.S.C. 103 rejection is based on 35 U.S.C. 102(a), 102(b), 102(e), etc. depending on the type of prior art reference used and its publication or issue date." MPEP § 2141.01.

To establish a *prima facie* case of obviousness, an Examiner must show that an invention would have been obvious to a person of ordinary skill in the art at the time of the invention. MPEP § 2141. "Obviousness is a question of law based on underlying factual inquiries." *Id.* The factual inquiries enunciated by the Court include "ascertaining the differences between the claimed invention and the prior art" and "resolving the level of ordinary skill in the pertinent art." MPEP § 2141.

"A statement that modifications of the prior art to meet the claimed invention would have been 'well within the ordinary skill of the art at the time the claimed invention was made' because the references relied upon teach that all aspects of the claimed invention were individually known in the art is not sufficient to establish a *prima facie* case of obviousness without some objective reason to combine the teachings of the references." MPEP § 2143.01. "[R]jections on obviousness cannot be sustained by mere conclusory statements; instead, **there must be some**

**articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."** MPEP § 2143.01 (citing *KSR*, 550 U.S. at \_\_\_, 82 USPQ2d at 1396) (emphasis added).

Moreover, "[i]f the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." MPEP § 2143.01. Also, "the proposed modification cannot render the prior art unsatisfactory for its intended purpose." MPEP § 2143.01.

**B. Crook May Not Be Properly Combined With Washington et al.**

The cited combination of Washington et al. and Crook is improper and, therefore, the pending claims are allowable over the cited combination of art. MPEP § 2143.01.

Washington et al. disclose a programmatically generated graphical program that may perform an instrumentation function, such as a test and measurement function, or an industrial automation function, as may be seen in Figures 2A and 2B (Washington et al., Col. 9, lines 39-43). One example of such a program is the LabVIEW graphical program. (Washington et al., Col. 8, lines 55-59). The system disclosed by Washington et al. is configured to generate graphs, such as the graphs shown in Figures 20, 21, 31 and 33 in response to measurement input received from other devices. (Washington et al., Col. 9, lines 50-62; Col. 10, lines 48-66; Col. 12, lines 15-23; Col. 26, lines 62-67; Col. 31, lines 30-46).

For the Examiner's reference, Applicant has attached hereto as Exhibit A copies of pages from the website operated by the maker of the LabVIEW program and the assignee of the Washington et al. patent. These pages were obtained from the website, [www.ni.com/labview](http://www.ni.com/labview).

In contrast, Crook discloses a system for configuring among call processing applications. (Crook, Abstract). The call processing system cannot be combined with the graphical generation system disclosed by Washington et al. First, such a combination would render Washington et al. inoperable for its intended purpose, generating graphical source code for the generation of graphs that illustrate input received from test modules, control monitoring devices, or other input/output devices or illustrating an analysis of such inputs via one or more of such graphs. (Washington et al., Col. 1, lines 15-22).

Second, such a combination would impermissibly change the principle of operation of the invention disclosed by Washington et al. A user of Crook's invention must create images by connecting certain images to others by arrows (See Crook at Figures 4-6 and Col. 6, lines 8-53) to generate graphical interface images of connection. In contrast, the system disclosed by Washington et al. is configured for a program to generate the graphical images and display those images based upon measurement input or test set-up data received by the program. Users are not permitted to create the graphical images, such as the graphs shown in Figures 32-33 of Washington et al., by drop and drag techniques. To the contrary, such graphs are generated by Washington et al.'s program based on input received from input received by a user's interaction with a "wizard" application and by input received from other devices. (Washington et al., Col. 16, lines 41-58; Col. 31, lines 20-50).

Third, the cited rationale for combining Crook and Washington et al. is illogical. The Examiner claims that one of ordinary skill in the art would be motivated to combine Crook and Washington et al. "in order to programmatically generate a graphical code to implement the specified functionality." (Office Action, at 4). However, the invention of Crook does not

programmatically generate a graphical code. To the contrary, images are updated based on a user using a mouse to point and drag or point and click on a graphical interface to generate connecting arrows. (Crook, Col. 6, lines 48-52). Washington et al., in contrast, requires their program to automatically, or programmatically, generate a graphical image based on a test set up or arrangement of nodes entered by a user via a "wizard." (Washington et al., Col. 16, lines 41-58). The use of such a "wizard" with the invention disclosed by Crook would not permit the functionality required by Crook. Indeed, the use of such a "wizard" clearly teaches away from Crook's invention.

**C. Crook And Washington et al. Do Not Disclose Each And Every Limitation of the Claims**

Even if Crook and Washington et al. are impermissibly combined, this combination fails to teach each and every limitation of the pending claims. Therefore, the pending claims are allowable over the cited combination of art.

- 1. Crook does not disclose "moving the element to one of the plurality of selectable instructions such that the one of the plurality of selectable instructions is a selected instruction" required in claims 9-16 or the "selection mechanism for moving the element to a selected instruction of the selectable instructions" of claims 17-23**

The Examiner contends that Crook discloses "moving the element to one of the plurality of selectable instructions, whereby the one of the plurality of selectable instructions is a selected instruction" and "a selection mechanism for moving the element to a selected an instance [sic] of the selectable instructions." (Office Action, at 3 & 6). The Examiner cites Column 4, lines 42-50 and 66-67 of Crook as disclosing such a feature. (*Id.*). However, the "drag and drop" reference of Column 4 cited in the Office Action relates to the selection of applications and locating those applications in a file window as shown in Figure 4. (Crook, Col. 6, lines 8-53).

The "drag and dropping" does not relate to the movement of elements to any selectable instructions.

Indeed, Crook teaches that a user must assign a selectable instruction by dragging and dropping files in a file window and, thereafter, configure the applications by using a pull-down configuration menu 63 and forming graphical connectors 68 by clicking on a connector 68 and dragging it from one application to another application. (Crook, Col. 6, lines 8-53; Figures 4-6). Crook does not disclose moving the element to one of the plurality of selectable instructions such that one of the plurality of selectable instructions is a selected instruction." To the contrary, Crook requires that a configuration menu 63 be selected, as shown in Figure 5 to configure an application and that a connector be drawn on a display to demonstrate a flow of a call. (Crook, Col. 6, lines 8-53; Figures 4-6).

The Examiner admits that Washington et al. fail to teach the moving elements or selection mechanism for moving elements required by the pending claims (Office Action at 3 & 6). However, as discussed above, Crook also does not disclose such a requirement. Clearly, the combination of Crook and Washington et al. do not teach each and every limitation of the pending claims. Therefore, the rejection of the pending claims should be withdrawn.

**D. Dependent Claims Are Also Independently Allowable Over The Cited Art**

Certain dependent claims are also independently allowable over the cited art. These dependent claims are discussed in more detail below.

**1. Claims 22-23 are allowable over the cited art**

The combination of art cited by the Examiner fails to teach or suggest each and every limitation of claims 22 and 23 as required to render the claims obvious. Therefore, these claims are allowable over the cited combination of art.

**a. The examiner's own evaluation of Washington et al. show that claims 22-23 are allowable**

At page 3 of the Office Action, the Examiner correctly stated that Washington et al. do not disclose "displaying a communication address via a movable element on the graphical user interface" nor "moving the element to one of the plurality of selectable instructions." The Examiner also correctly found that "Washington et al. does not disclose "a selection mechanism for moving the element to a selected an instance [sic] of the selectable instructions." (Office Action at 6). However, at page 7 of the Office Action, the Examiner says that Washington et al. disclose repeating the moving of an element and that the combination between an address and selected instruction is canceled as a result of the repeated movement at Figure 15 and Column 25, lines 31-35 to reject claims 22 and 23. This is incorrect. Washington et al. do not teach or suggest any selection mechanism as required by Applicant's claims 22 and 23.

**b. Washington et al. do not disclose a selection mechanism configured to repeat the moving of any element**

Applicant's claims 22 and 23 also require a selection mechanism to repeatedly move an element to a selected instruction. As the Examiner has acknowledged, Washington et al. do not

disclose any selection mechanism for "moving the element to a selected instruction of the selectable instructions." Therefore, Washington et al. also do not disclose a "selection mechanism is configured to repeat the moving of the element" as required by claims 22 and 23.

**c. Washington et al. do not disclose the cancellation of the combination between the at least one communication address and the selected instruction as a result of the repeated movement of the element**

Washington et al. also do not disclose the cancellation of the combination between at least one communication address and a selected instruction as a result of the repeated movement of the movable element to the selected instruction. As correctly acknowledged by the Examiner, Washington et al. does not disclose any movement of an element by a selection mechanism. (Office Action, at 3 and 6). The Examiner cites to Figure 15 and Column 25, lines 31-35 of Washington et al. as disclosing a repeated movement of an element as cancelling a combination of a communication address and a selected instruction. (Office Action, at 7). However, Washington et al. do not teach such limitations at Figure 15 or Column 25. Indeed, these portions are silent with respect to these limitations. As the Examiner has correctly acknowledged, Washington et al. do not teach such limitations. (Office Action, at 3 & 6). Indeed, this portion of Washington et al. only generally discusses the replacement of graphical source code for a particular node and does not disclose any repeated movement of elements or the cancellation of a combination based on the repeated movement of any elements.

**2. Claims 12 and 16 are allowable over the cited art**

The Examiner contends that Washington et al. discloses repeated movement of elements as required by claims 12 and 16. (Office Action, at 4-5). As discussed above with reference to



claims 22 and 23, Washington et al. do not teach or suggest any repeated movement of any elements. Claims 12 and 16 are also clearly allowable over the cited combination of art.

### **III. CONCLUSION**

For at least the above reasons, reconsideration and allowance of all pending claims are respectfully requested.

Respectfully submitted,

/Ralph G. Fischer/

Dated: April 30, 2009

(412) 392-2121

Ralph G. Fischer  
Registration No. 55,179  
BUCHANAN INGERSOLL & ROONEY PC  
301 Grant Street , 20th Floor  
Pittsburgh, Pennsylvania 15219  
Attorney for Applicant

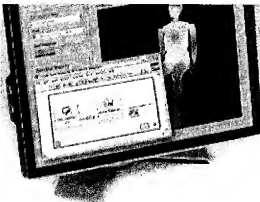
## **EXHIBIT A**



Improve your ni.com experience. Login or Create a user profile.  
Questions? Call (800) 531-5066

## NI LabVIEW

What's New in  
LabVIEW 8.6  
What Is  
LabVIEW?  
Download  
LabVIEW  
Add-Ons and  
Toolkits  
LabVIEW Zone  
Community  
Feature  
Comparison



## Introducing LabVIEW 8.6

Go Parallel with the Latest Wi-Fi,  
Multicore, and FPGA Technologies

» See what's new

For more than 20 years, NI LabVIEW graphical programming has revolutionized the development of test, measurement, and control applications. Regardless of experience, engineers and scientists can rapidly and cost-effectively interface with measurement and control hardware, analyze data, share results, and distribute systems.

Graphical Programming

High-Level Development Tools

Built-In Measurement and Analysis Functions

Multiplatform and Embedded Devices

**LabVIEW Development Systems - Available for Windows | Mac OS | Linux® OS**

Base :

**\$1,249**

Graphical UI development

Data acquisition

Instrument control

Reporting and file I/O

Purchase | Upgrade | Compare

Full :

**\$2,599**

More than 600 math/analysis functions

External code integration

Web connectivity

Advanced UI development

Purchase | Upgrade | Compare

Professional :

**\$4,299**

Application distribution

Development management

Source code control

Network communication

Purchase | Upgrade | Compare  
NI Developer Suite >

**\$4,699**

Best value for software  
Custom software selection  
Included add-ons and toolkits  
Quarterly software updates

Purchase | Upgrade | Compare

Attend the 2009 Virtual Automated Test Summit

Get free industry training on applying best practices in test



See the agenda  
Coming Soon: LabVIEW 2009

Test the Next Version of LabVIEW: Join the Beta Program



Sign up now  
Software Engineering and Validation

Validate your applications with NI LabVIEW Software Engineering Tools



Learn more

#### Customer Successes

Microsoft Maximizes Xbox 360 Test Coverage  
NASA controls MEMS microshutters on  
advanced telescope

Woodward Governor Develops Engine Simulator  
I<sup>2</sup>S Steel Rolling Mills Gain Efficiency and  
Accuracy  
More ...

#### LabVIEW Add-On Modules and Toolkits

Embedded Design  
Control Design and Simulation  
Image and Signal Processing

Industrial Monitoring and Control  
Software Development and Deployment  
Report Generation and Data Storage

#### LabVIEW Community



NI LabVIEW Zone

Example Code

#### Development Resources

Technical Support  
Training and Certification  
Instrument Driver Network  
Tools Network

[Discussion Forums](#)  
[User Groups](#)  
[Learning Center](#)

## LabVIEW in Academia

Just as LabVIEW is revolutionizing industry, it is dramatically affecting traditional academic research and teaching. A laboratory based on LabVIEW makes researchers more productive and improves the way students learn. Rather than focusing on sometimes-tedious methods of gathering data, educators and students can focus on results and concepts.

[Why LabVIEW in Education?](#) | [LabVIEW Student Edition](#) | [Academic Discounts](#)

Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

## Assistance and Contact



**Questions? Get real-time assistance now!**

» [Global Contact Information](#)

[My Profile](#) | [RSS](#) | [Privacy](#) | [Legal](#) | [Contact NI](#)

© 2009 National Instruments Corporation. All rights reserved.

| ☒ [E-Mail this Page](#)



United States

Automated Test and  
Instrument Control

Data Acquisition

Industrial Measurements  
and Control

Embedded Design

Academia



### Increase Throughput with Parallel Test

Learn how test engineers use LabVIEW, multicore processors, and new bus technologies to create high-performance test systems capable of parallel processing, parallel measurements, and even parallel test on the production floor.

[Read the white paper](#)

### Connect to Any Instrument, Any Sensor, Any Bus

Read how built-in I/O and communication libraries in LabVIEW provide native connectivity to any instrument, sensor, bus, or software interface to simplify integration of these components into your test applications.

[Read the white paper](#)

### Boeing Uses LabVIEW to Develop a Low-Cost Test System

See how LabVIEW software and NI hardware helped a single Boeing developer create a high-channel-count, synchronized test system in only six months to measure the effectiveness of new commercial jetliner designs in reducing noise during flight.

[Watch the video](#)

Ready to take the next step?

[▶ Try LabVIEW](#)

Acquire Measurements from Any



### Sensor, Any Bus

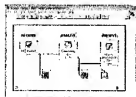
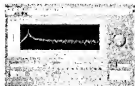
Use LabVIEW to create a fully functional measurement application with analysis and a custom user interface using a variety of PCI- and USB-based data acquisition hardware.

[View the webcast](#)

### Measure in Minutes with LabVIEW and the DAQ Assistant

Read how LabVIEW uses the interactive DAQ Assistant and high-level functions to combine the flexibility and scalability of traditional programming languages and the ease of use of configuration-based data acquisition tools.

[Read the white paper](#)



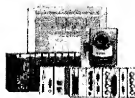
### Acquire, Analyze, and Present Data Quickly with Express VIs

Learn how to develop a powerful DAQ application that includes advanced analysis and a custom user interface. See how tasks that would take several lines of code in traditional programming languages are interactively configured with Express VIs in LabVIEW.

[Watch the video](#)

Ready to take the next step?

[▶ Try LabVIEW](#)



### Use LabVIEW to Program the Next-Generation PLC

Learn how industrial engineers pushing the boundaries of controller technology can use LabVIEW graphical programming and programmable automation controllers (PACs) to combine PC functionality with programmable logic controller (PLC) reliability.

[Read the white paper](#)

### Add Advanced Analysis to Your PLC

Add advanced analysis, signal processing, decision making, and debugging diagnostics to an existing PLC-based industrial application with LabVIEW and OPC connectivity.

[View the webcast](#)





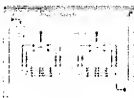
### Control Industrial Machinery Remotely with LabVIEW

See how Nexans uses LabVIEW and NI reconfigurable embedded hardware to control the hydraulic systems on a remotely operated underwater excavator that prepares the ocean floor for a pipeline to extract natural gas.

[Watch the video](#)

Ready to take the next step?

[▶ Try LabVIEW](#)



### Simplify Embedded Development with Graphical System Design

Discover how LabVIEW graphical system design software provides domain experts with high-level tools, such as statecharts, to design and implement their systems on off-the-shelf hardware.

[View the webcast](#)

### Get to Market Faster with LabVIEW and COTS Hardware

Read how LabVIEW graphical programming and commercial off-the-shelf (COTS) hardware help design teams get products to market faster by accelerating every stage of development - from the earliest stages of design and simulation to prototyping the system with real-world signals and deploying to a chosen processor target.

[Read the white paper](#)



### Prototype and Deploy a Custom Controller with LabVIEW

See how Driven used LabVIEW and COTS prototyping hardware to quickly develop custom IP for an FPGA-based engine control unit (ECU) in a high-performance motorcycle engine.

[See the video](#)

Ready to take the next step?

[▶ Try LabVIEW](#)

Combine Graphical and Textual  
Programming to Reduce Design Time





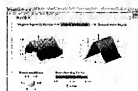
Reduce embedded design time by using a LabVIEW graphical system design approach to combine the traditionally separate tasks of theoretical design and prototyping. Choose between graphical and textual programming throughout the process.

[View the webcast](#)

### Choose the Software Preferred by Students for Signal Processing

Read how Professor Mark Yoder, Ph.D., recently transitioned the signal processing course at Rose-Hulman from The MathWorks, Inc. MATLAB® software to LabVIEW software. Dr. Yoder's research later showed that students prefer LabVIEW as a learning tool by a 3 to 1 margin. MATLAB® is a registered trademark of The MathWorks, Inc.

[Read the white paper](#)



### Watch Students Use LabVIEW to Create Segway-Inspired Machine

See how a senior design team at Ransselaar Polytechnic Instituut used LabVIEW to develop a two-wheeled robotic locomotion platform inspired by the Segway Human Transporter. With LabVIEW software and NI hardware, the students could use one platform throughout the project.

[Watch the video](#)

Ready to take the next step?

[▶ Try LabVIEW](#)